IEICE TRANS. INF. & SYST., VOL.E96–D, NO.3 MARCH 2013

PAPER *Special Section on Foundations of Computer Science*

Reconstruction Algorithms for Permutation Graphs and Distance-Hereditary Graphs*

Masashi KIYOMI[†a)], **Toshiki SAITOH**[††], ***Nonmembers*, *and* Ryuhei UEHARA**[†††], ***Member***

SUMMARY PREIMAGE CONSTRUCTION problem by Kratsch and Hemaspaandra naturally arose from the famous graph reconstruction conjecture. It deals with the algorithmic aspects of the conjecture. We present an $O(n^8)$ time algorithm for PREIMAGE CONSTRUCTION on permutation graphs and an $O(n^4(n+m))$ time algorithm for PREIMAGE CONSTRUCTION on distance-hereditary graphs, where n is the number of graphs in the input, and m is the number of edges in a preimage. Since each graph of the input has $n-1$ vertices and $O(n^2)$ edges, the input size is $O(n^3)$ (, or $O(nm)$). There are polynomial time isomorphism algorithms for permutation graphs and distance-hereditary graphs. However the number of permutation (distance-hereditary) graphs obtained by adding a vertex to a permutation (distance-hereditary) graph is generally exponentially large. Thus exhaustive checking of these graphs does not achieve any polynomial time algorithm. Therefore reducing the number of preimage candidates is the key point.

key words: *the graph reconstruction conjecture, permutation graphs, polynomial time algorithm*

1. Introduction

The graph reconstruction conjecture proposed by Ulam and Kelly** has been studied by many researchers intensively. We call the multi-set $\{G_1, G_2, \dots, G_n\}$ the *deck* of a graph $G = (V{=}\{v_1, v_2, \dots, v_n\}, E)$ if G_i is isomorphic to $G - v_i$ for every $i \in \{1, 2, \dots, n\}$, where $G - v$ is a graph obtained from G by removing v and incident edges. A graph G is a *preimage* of a deck of a graph G' if G and G' has the same deck. We also say that a graph G is a preimage of the n graphs when they are the deck of G. The graph reconstruction conjecture is that there is at most one preimage of given n graphs ($n \geq 3$). No one has given a proof nor a counter example of this conjecture, while small graphs are verified positively [15].

Kelly's Lemma [11] is well-known, and is a basic tool. It shows that, let G be any preimage of the given deck, and let H be a graph whose number of vertices is smaller than that of G. Then we can uniquely determine the number of subgraphs in G isomorphic to H from the deck. Greenwell and Hemminger extended this lemma to a more general form [8]. We can know the degree sequence of a preimage from these lemmas. Kelly also showed that the conjecture is true on regular graphs, trees, and disconnected graphs. Tutte proved that the dichromatic rank and Tutte polynomials are reconstructible (i.e., looking at the deck, they are uniquely determined) [20]. Bollobás showed that almost all graphs are reconstructible from three well-chosen graphs in its deck [2]. About permutation graphs, Rimscha showed that permutation graphs are recognizable in the sense that looking at the deck of G one can decide whether or not G belongs to permutation graphs [21]. To be precise Rimscha showed in the paper that comparability graphs are recognizable. Even's result [6] directly gives a proof in the case of permutation graphs. Rimscha also showed in the same paper that many subclasses of perfect graphs including perfect graphs themselves are recognizable, and some of subclasses are reconstructible. There are a lot of papers about the conjecture, and many good surveys about this conjecture. See for example [3], [9].

There are several kinds of algorithmic problems related to the graph reconstruction conjecture. We consider algorithmic problems proposed by Kratsch and Hemaspaandra [13] described below.

- Given a graph G and a multi-set of graphs D, check whether D is the deck of G (DECK CHECKING).
- Given a multi-set of graphs D, determine whether there is a graph whose deck is D (LEGITIMATE DECK).
- Given a multi-set of graphs D, construct a graph whose deck is D (PREIMAGE CONSTRUCTION).
- Given a multi-set of graphs D, compute the number of (pairwise nonisomorphic) graphs whose decks are D (PREIMAGE COUNTING).

Kratsch and Hemaspaandra showed that these problems are solvable in polynomial time for graphs of bounded degree, partial k-trees for any fixed k, and graphs of bounded genus, in particular for planar graphs [13]. In the same paper they proved many GI related complexity results. Hemaspaandra et al. extended the results [10]. The authors presented a polynomial time PREIMAGE CONSTRUCTION algorithm for interval graphs [12].

We present an $O(n^8)$ time algorithm for PREIMAGE CONSTRUCTION on permutation graphs, and an

Manuscript received March 28, 2012.
Manuscript revised June 28, 2012.
†The author is with International College of Arts and Sciences, Yokohama City University, Yokohama-shi, 236–0027 Japan.
††The author is with the Department of Electrical and Electronic Engineering, Graduate School of Engineering, Kobe University, Kobe-shi, 657–8501 Japan.
†††The author is with School of Information Science, Nomi-shi, 923–1292 Japan.
*A preliminary version of this article was presented at WALCOM 2010.
a) E-mail: masashi.kiyomi@gmail.com
DOI: 10.1587/transinf.E96.D.426

**Determining the first person who proposed the graph reconstruction conjecture is difficult, actually. See [9] for the detail.

Copyright © 2013 The Institute of Electronics, Information and Communication Engineers

$O(n^4(n+m))$ time algorithm for PREIMAGE CONSTRUCTION on distance-hereditary graphs, where m is the number of edges in a preimage[†]. Since permutation graphs and distance-hereditary graphs have characterizations by forbidden graphs, it is easy to see that every graph in the deck of a permutation (distance-hereditary) graph is a permutation (generalized distance-hereditary) graph. Note that while a distance-hereditary graph is connected, some graphs obtained by removing a vertex from it are not necessarily connected, therefore, the deck contains generalized distance-hereditary graphs[††]. We propose PREIMAGE CONSTRUCTION algorithm for a deck consisting of permutation (generalized distance-hereditary) graphs. We state our main theorems below.

Theorem 1: There is an $O(n^8)$ time PREIMAGE CONSTRUCTION algorithm for a deck D consisting of n permutation graphs.

Theorem 2: There is an $O(n^4(n+m))$ time PREIMAGE CONSTRUCTION algorithm for a deck D consisting of n generalized distance-hereditary graphs.

2. Preliminaries

2.1 Notations

All the graphs in this paper are simple. We denote the complement of graph G by $\overline{G}$.

Let $G = (V, E)$ be a graph, and let $V' \subset V$ is a vertex subset of G. We denote by $G[V']$ the graph induced by V' from G.

We denote by $N_G(v)$ the neighbor set of vertex v, and by $N_G[v]$ the closed neighbor set of vertex v in graph G. "Closed" means that $N_G[v]$ contains v itself. Vertices u and v are called *strong twins* if $N_G[u]$ is equal to $N_G[v]$, and *weak twins* if $N_G(u)$ is equal to $N_G(v)$.

Let S be a set, and $s \in S$. We denote $S \setminus \{s\}$ by $S - s$.

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ on the disjoint vertex sets ($V_1 \cap V_2 = \emptyset$), graph $G = (V_1 \cup V_2, E_1 \cup E_2)$ is a disjoint union of G_1 and G_2. Disjoint union of three or more graphs is defined in the analogous way.

We now define graph $H(2n)$. $H(2n)$ is a bipartite graph (X, Y, E) such that $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$, and $\{x_i, y_j\} \in E$ iff $i \leq j$. See Fig. 1.

Let $\pi = (\pi_1, \ldots, \pi_n)$ be a permutation of $1, \ldots, n$. A permutation diagram of π is a set of n line segments $l_1, \ldots, l_n$ that connect two parallel lines L_1, L_2 on Euclidean plane

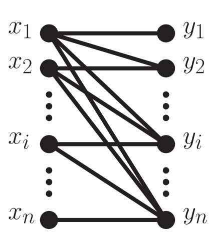

Fig. 1 Graph $H(2n)$.

such that end-points of $l_1, \ldots, l_n$ appear in this order on L_1, and appear in the order of $\pi_1, \ldots, \pi_n$ on L_2. A permutation diagram defines a permutation in a natural way. See Fig. 2. We denote by π^{V} the permutation whose permutation diagram is obtained by reversing that of π vertically, by π^{H} the permutation whose permutation diagram is obtained by reversing that of π horizontally, and by π^{R} the permutation whose permutation diagram is obtained by reversing that of π both vertically and horizontally[†††].

2.2 Permutation Graphs

Let π be a permutation of the numbers $1, 2, \ldots, n$. $G(\pi) = (V, E)$ is a graph satisfying that

- $V = \{1, \ldots, n\}$, and
- $\{i, j\} \in E \Leftrightarrow (i - j)(\pi_i^{-1} - \pi_j^{-1}) < 0$.

A graph G is called a *permutation graph* if there exists a permutation π such that G is isomorphic to $G(\pi)$. Equivalently a graph G is a permutation graph if there exists a permutation π such that G is an intersection model of the permutation diagram of π.

A graph G is a permutation graph if and only if both G and its complement $\overline{G}$ are comparability graphs [6]. Thus if a graph G is a permutation graph, $\overline{G}$ is also a permutation graph.

For two permutation graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ satisfying $|V_1| = |V_2| = n$, there is an $O(n^2)$ time algorithm that determines if G_1 and G_2 are isomorphic [18].

Gallai characterized comparability graphs with the forbidden subgraphs [7]. Since permutation graphs are equivalent to comparability and co-comparability graphs [6], the characterization of permutation graphs is easily obtained. A graph G is a permutation graph if and only if G is (C_{k+6}, T_2, X_2, X_3, X_{30}, X_{31}, X_{32}, X_{33}, X_{34}, X_{36}, XF_1^{2k+3}, XF_2^{k+1}, XF_3^{k}, XF_4^{k}, XF_5^{2k+3}, XF_6^{2k+2}, co-C_{k+6}, co-T_2, co-X_2, co-X_3, co-X_{30}, co-X_{31}, co-X_{32}, co-X_{33}, co-X_{34}, co-X_{36}, co-XF_1^{2k+3}, co-XF_2^{k+1}, co-XF_3^{k}, co-XF_4^{k}, co-XF_5^{2k+3}, co-XF_6^{2k+2}, and odd-hole)-free. See Fig. 3.

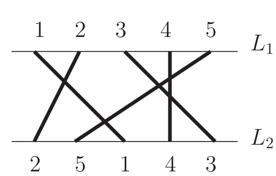

Fig. 2 The permutation diagram of permutation $(2, 5, 1, 4, 3)$.

[†]The number of edges in a preimage is reconstructible for any graphs. See [11]

[††]We define generalized distance-hereditary graphs later. They are intuitively not necessarily connected version of distance-hereditary graphs.

[†††]For those who want concrete description, it is easy to see that $\pi^{\mathrm{V}} = \pi^{-1}$ holds. Using a notion $\overline{\pi} = (\pi_n, \ldots, \pi_1)$, $\pi^{\mathrm{H}} = \overline{\overline{\pi}^{-1}}^{-1}$, and $\pi^{\mathrm{R}} = \overline{\pi}^{-1}$ also hold.

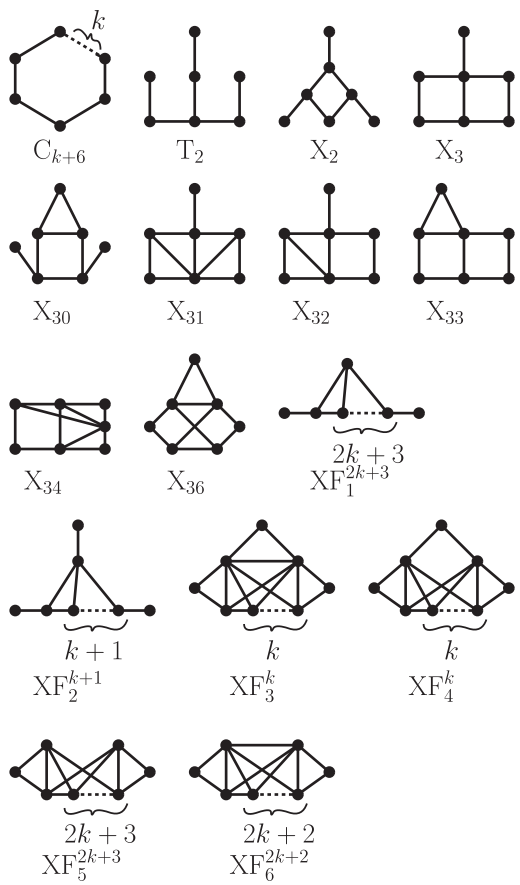

Fig. 3 Forbidden graphs of permutation graphs are these graphs ($k \geq 0$), the complements of them, and odd-holes.

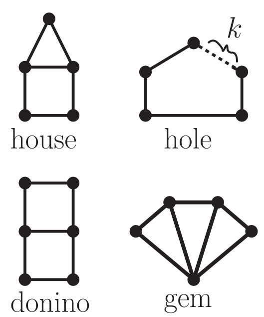

Fig. 4 Forbidden graphs of distance hereditary graphs ($k \geq 0$).

2.3 Distance-Hereditary Graphs

A *distance-hereditary* graph $G = (V, E)$ is a connected graph such that, in any connected induced subgraph H of G, any pair of vertices u and v in H has the same distance as in G.

We can check if the given two distance-hereditary graphs are isomorphic in $O(n+m)$ time, where n is the number of the vertices and m is the number of edges [16].

It is known that a connected graph G is distance-hereditary if and only if G is (house, hole, domino and gem)-free [1]. See Fig. 4.

One of the good properties of distance-hereditary graphs is that a distance-hereditary graph can be generated from a single vertex by the following operations:

(a) Add a new vertex and connect it to one vertex in the graph by an edge.
(b) Copy a vertex so that the new vertex and the original one are weak twins.
(c) Copy a vertex and connect the new vertex and the original one by an edge so that they are strong twins.

In fact, this is another characterization of distance-hereditary graphs [1].

Thinking not necessarily connected version of distance-hereditary graphs is sometimes convenient. Thus, we define such a graph class. A graph G is *generalized distance-hereditary graph* if G is a disjoint union of distance-hereditary graphs. It is easy to see that a generalized distance-hereditary graphs is a graph not containing house, hole, domino, and gem as an induced subgraph.

2.4 Modular Decomposition

Modular decomposition is a strong tool for developing fast algorithms in many areas. Here we summarize it. For the detail see for example [4], [19].

Let $G = (V, E)$ be a graph. The subset $M \subset V$ is a *module* in G, if for all the vertices $u, v \in M$ and $w \in V \setminus M$, $\{u, w\} \in E$ if and only if $\{v, w\} \in E$. A module M in G is *trivial* if $M = V$, $M = \emptyset$, or $|M| = 1$. G is called a *prime* (with respect to modular decomposition) if G contains only trivial modules. A module M is *strong* if it does not overlap any other modules in G, i.e.,

$$M \cap M' = \emptyset,\ M \subset M',\ \text{or } M' \subset M \quad (\text{for } {}^{\forall}M' : \text{module in } G)$$

holds. We call a module that contains at least two vertices a *multi-vertex module*.

A *modular decomposition tree* of a graph G is a tree whose each node corresponds to each strong module of G such that for any two nodes N_1 and N_2 which correspond to modules M_1 and M_2 respectively, N_1 is an ancestor of N_2 if and only if M_1 contains M_2. We sometimes say that strong module M_1 is a parent of strong module M_2, and M_2 is a child of M_1, if the node corresponding to M_1 is the parent of the node corresponding to M_2 in the modular decomposition tree.

A strong multi-vertex module M in graph G such that the graph obtained from $G[M]$ by contracting every its child module to a vertex has no edge is *parallel module*. A strong multi-vertex module M in graph G such that the graph obtained from $G[M]$ by contracting every its child module to a vertex is a complete graph is *series module*. Let M' be a strong multi-vertex module. If M' is not a parallel module, and M' is not a series module, then M' is called a *prime module*. A graph induced by a prime module is connected in both G and $\overline{G}$ [19]. We show an example of a permutation

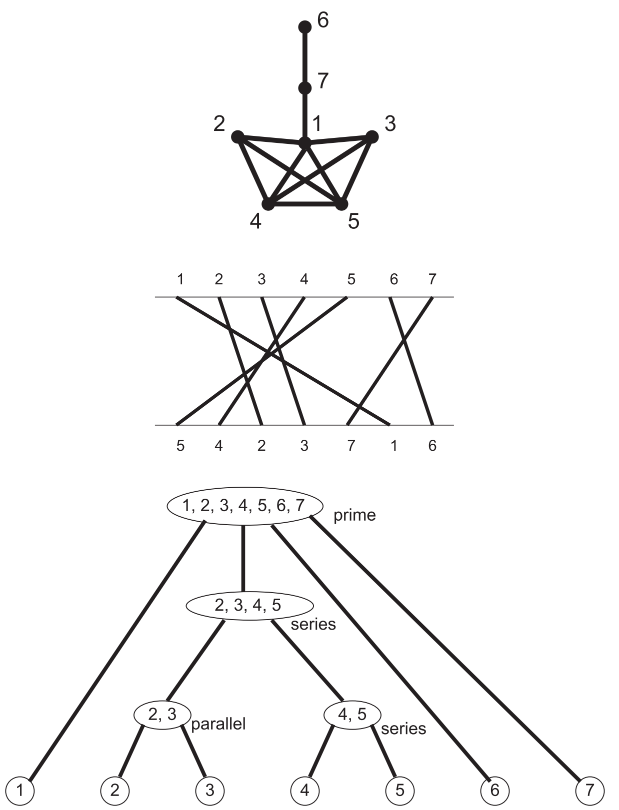

Fig. 5 An example of a permutation graph, its permutation diagram, and its modular decomposition tree.

graph and its modular decomposition tree in Fig. 5.

We say a strong multi-vertex module M is *minimal* if every child of M is a module of one vertex. Note that every graph of the size more than one has at least one minimal strong multi-vertex module. We introduce a basic lemma.

Lemma 3 (Gallai [7]): A minimal strong multi-vertex module that is a prime module induces a prime.

Let $G = (V, E)$ be a prime. We say that G is *critical* if $G - v$ is not a prime for any $v \in V$. It is known that a critical graph $G = (V, E)$ is isomorphic to $H(|V|)$ or to $\overline{H(|V|)}$ [17]. Hence the number of vertices in a critical graph is always even.

It is known that a permutation graph G that is a prime with respect to modular decomposition has a unique representation [4], [14]. Note that $G(\pi)$, $G(\pi^{\mathrm{V}})$, $G(\pi^{\mathrm{H}})$, and $G(\pi^{\mathrm{R}})$ are isomorphic. Thus the sentence "G has a unique representation" here means that there are at most four permutations $\pi, \pi^{\mathrm{V}}, \pi^{\mathrm{H}}$, and π^{R} whose representing graphs are isomorphic to G.

3. Polynomial Time Reconstruction Algorithm

Our algorithm outputs preimages that are permutation (distance-hereditary) graphs. However it is possible that a non-permutation (non-distance-hereditary) graph has a deck that consists of permutation (generalized distance-hereditary) graphs, though it is exceptional. Since considering this case all the time in the main algorithm makes it complex, we attempt to get done with this special case in Sect. 3.1.

Then we present DECK CHECKING algorithms for permutation graphs and distance-hereditary graphs. Since an $O(n^2)$ time isomorphism algorithm for permutation graphs [18] and a linear time one for distance-hereditary graphs [16] are known, developing polynomial time DECK CHECKING algorithms is not very difficult.

Next we present our main algorithms. We first show an algorithm for permutation graphs, and then show one for distance-hereditary graphs. Our main algorithm for permutation graphs has two parts. One is for a preimage G that has a minimal strong multi-vertex module M such that $G[M]$ is not critical, and the other part is for otherwise. In both the parts, we construct polynomially many candidates of a preimage, and use DECK CHECKING algorithm to check whether each candidate is a preimage. Since we of course do not know the properties of a preimage when we are given an input deck, we execute both these two parts for the input deck.

3.1 Non-permutation (Non-distance-hereditary) Graph Preimage Case

Let D be a deck consisting of n graphs $G_1, G_2, \ldots, G_n$. It is clear that $G_1, G_2, \ldots, G_n$ have the same number of vertices $n - 1$, and that the number of vertices in a preimage G is n. Note that, if G is not a permutation (distance-hereditary) graph though $G_1, G_2, \ldots, G_n$ are permutation (generalized distance-hereditary) graphs, G must be one of the forbidden graphs. Since the number of the forbidden graphs of the size n is $O(1)$, we can check if one of them is a preimage of the input graphs in the polynomial time with DECK CHECKING algorithm which we will describe in the next subsection. The time complexity is $O(n^4)$ for a deck consisting of permutation graphs, and $O(n^2(n + m))$ for a deck consisting of generalized distance-hereditary graphs, since the time complexity of the DECK CHECKING algorithm is $O(n^4)$ for permutation graphs and $O(n^2(n + m))$ for generalized distance-hereditary graphs, where m is the number of edges in G.

Theorem 4: If n permutation graphs $G_1, G_2, \ldots, G_n$ have a preimage G that is not a permutation graph, we can reconstruct G from $G_1, G_2, \ldots, G_n$ in $O(n^4)$ time.

Theorem 5: If n distance-hereditary graphs $G_1, G_2, \ldots, G_n$ have a preimage G that is not a distance-hereditary graph, we can reconstruct G from $G_1, G_2, \ldots, G_n$ in $O(n^2(n + m))$ time.

3.2 DECK CHECKING

Given a deck D that consists of permutation (generalized distance-hereditary) graphs, and given a preimage candidate

$G = (V, E)$ whose deck consists of permutation (generalized distance-hereditary) graphs, we first prepare the deck $\hat{D}$ of G in $O(|V|(|V|+|E|))$ time. Then we can check if D and $\hat{D}$ are the same by using the isomorphism algorithms in [18] and [16] $O(|V|^2)$ times. They costs $O(|V|^2 \cdot |V|^2)=O(|V|^4)$ time for permutation graphs, and $O(|V|^2(|V| + |E|))$ time for distance-hereditary graphs. Therefore we obtain the theorems below.

Theorem 6: There is $O(|V|^4)$ time DECK CHECKING algorithm for a deck that consists of permutation graphs, and a preimage candidate $G = (V, E)$ whose deck consists of permutation graphs.

Theorem 7: There is $O(|V|^2(|V| + |E|))$ time DECK CHECKING algorithm for a deck that consists of generalized distance-hereditary graphs, and a preimage candidate $G = (V, E)$ whose deck consists of generalized distance-hereditary graphs.

3.3 Non-critical Case for Permutation Graph PREIMAGE CONSTRUCTION

First we consider the case that a preimage $G = (V, E)$ has a minimal strong multi-vertex module M such that $|M| \geq 3$, and $G[M]$ is not critical. If M is a prime module, since $G[M]$ is a prime due to Lemma 3, $G[M]$ has a vertex v such that $G[M] - v$ is a prime, and hence $M - v$ is a minimal strong multi-vertex module of $G[M] - v$. If M is not a prime module, due to the definition of modular decomposition, $G[M]$ is a complete graph, or $G[M]$ consists of isolated vertices. And thus $G[M]$ also has a vertex v such that $M - v$ is a minimal strong multi-vertex module of $G[M] - v$.

We search for a preimage by adding a vertex v to every minimal strong multi-vertex module M' of every graph in the deck to check if M' is the desired $M - v$. For every candidate, we use the DECK CHECKING algorithm to check if it is a preimage.

If we can specify $N_G(v)$, we can construct a candidate of G. We can easily specify $N_G(v) \setminus M'$, since $M' \cup \{v\}$ should be a module in G, i.e., every vertex in M' and v should seem the same from the vertices in $V \setminus M'$. Thus the remaining task is to specify $N_G(v) \cap M'$.

Due to the definition of a modular decomposition, M' is one of a clique, a collection of isolated vertices, and a module that induces a prime. It is not difficult to construct the candidate of G if M' is a clique, or M' consists of isolated vertices, since we know the degree sequence of G,† that is, we know the degree $\deg_G(v)$ of v in G. To be concrete, we have to connect v to $\deg_G(v) - |N_G(v) \setminus M'|$ vertices in M'.

Next we consider the case that $G[M']$ is a prime. A permutation graph that is a prime with respect to modular decomposition has a unique representation [4], [14]. Thus there are only $O(n^2)$ ways of connection of v and vertices in M'. Note that the number of permutation diagrams obtained by adding a line segment to a permutation diagram is clearly $O(n^2)$, since there are $O(n)$ choices for the end-point on L_1, and there are $O(n)$ choices for the end-point on L_2.

```
for each graph G' in the deck {
    for each minimal strong multi-vertex module M' of G' {
        prepare a isolated vertex v;
        connect v to vertices in V \ M' suitably;
        if M' is a clique, or M' are isolated vertices {
            connect v to deg_G(v) − |N_G(v) \ M'| vertices in M';
            do DECK CHECKING;
        } else {
            create a unique permutation diagram of G[M'];
            for each way of adding v
                do DECK CHECKING;
        }
    }
}
```

Fig. 6 The algorithm for the case that a preimage has a module that does not induce a critical graph.

Therefore by checking each of $O(n^2)$ candidates whether it is a preimage with the DECK CHECKING algorithm, we have a polynomial time algorithm. We show in Fig. 6 the whole algorithm for the case that a preimage has a module that does not induce a critical graph.

We now mention the time complexity of the algorithm in Fig. 6. There are n graphs in the deck. Each graph in the deck has $O(n)$ minimal strong multi-vertex modules. We can compute these modules in $O(n+m)$ time [5]. The time complexity of DECK CHECKING is $O(n^4)$. We can compute a permutation diagram of a permutation graph in $O(n + m)$ time. Therefore the time complexity of the algorithm is $O(n \cdot n((n + m) + n^2 \cdot n^4)) = O(n^8)$. Hence we have the theorem below.

Theorem 8: If a preimage $G = (V, E)$ that is a permutation graph has a minimal strong multi-vertex module M such that $|M| \geq 3$, and $G[M]$ is not critical, we can reconstruct G in $O(n^8)$ time.

3.4 Critical Case for Permutation PREIMAGE CONSTRUCTION

Lastly we consider the case that for every minimal strong multi-vertex module M of a preimage $G = (V, E)$, $G[M]$ is critical, or every minimal strong multi-vertex module has the size two.

Assume that all the minimal strong multi-vertex modules of G have the size two. Since a module of the size two makes twins, the reconstruction of G is easy in this case. Any graph G' in the deck is obtained by removing a vertex that is one of twins from G. Thus G can be reconstructed by copying a vertex in G'. We make weak and strong twins of each vertex of every graph in the deck, and check whether the obtained graph is a preimage by the DECK CHECKING algorithm. This achieve a polynomial time algorithm.

Now we consider the case that some of minimal strong multi-vertex modules in G have the size more than two. Let M be a minimal strong multi-vertex module of G whose

†Kelly's lemma directly gives the degree sequence of a preimage. See [11].

```
for each graph G' in the deck {
    for each vertex v of G' {
        make weak twin v' of vertex v;
        do DECK CHECKING;
        for each edge e incident to v' {
            remove e;
            do DECK CHECKING;
            add e;
        }
        remove v';
        make strong twin v' of vertex v;
        do DECK CHECKING;
        for each edge e incident to v' {
            remove e;
            do DECK CHECHEN;
            add e;
        }
        remove v';
    }
}
```

Fig. 7 The algorithm for the case that a preimage has a module that induces a critical graph.

size is more than two. Then since $\overline{G[M]}$ is a critical graph, $G[M]$ is isomorphic to $H(|M|)$ or $\overline{H(|M|)}$ (Fig. 1). The vertices x_1 and x_2 are almost twins in both the $H(|M|)$ and $\overline{H(|M|)}$. In fact $N_{H(|M|)}(x_1)$ and $N_{H(|M|)}(x_2)$ differ only in y_1, and $N_{\overline{H(|M|)}}[x_1]$ and $N_{\overline{H(|M|)}}[x_2]$ also differ only in y_1. We denote by v_1 and v_2 the vertices in M corresponding to x_1 and x_2 such that $|N_{G[M]}(v_1)| = |N_{G[M]}(v_2)| + 1$, or $|N_{G[M]}[v_1]| = |N_{G[M]}[v_2]| + 1$ holds. Since M is a module of G, $N_G(v_1)$ contains exactly one vertex in addition to the vertices in $N_G(v_2)$, or $N_G[v_1]$ contains exactly one vertex in addition to the vertices in $N_G[v_2]$.

Now we consider $G - v_2$. $G - v_2$ must be in the deck. Thus we check for every graph G' in the deck if it is $G - v_2$. If G' is $G - v_2$, we can reconstruct G from G' by copying a vertex in G' and removing an edge. We show the algorithm in Fig. 7.

We now focus on the time complexity. There are $O(n)$ graphs in the deck. The number of vertices in each graph is $O(n)$. We have to remove $O(n)$ edges in each iteration. The time complexity of DECK CHECKING is $O(n^4)$. Thus the total time complexity of the algorithm is $O(n \cdot n \cdot n \cdot n^4) = O(n^7)$. Thus we have the theorem below.

Theorem 9: If every minimal strong multi-vertex module of a graph G induces a critical graph, or if every minimal strong multi-vertex module of a graph G has the size two, we can reconstruct G in $O(n^7)$ time.

Combining Theorem 4, 8, and 9, we have the Theorem 1.

3.5 Distance-Hereditary Graph PREIMAGE CONSTRUCTION

Distance-hereditary graph $G = (V, E)$ can be constructed either (a) adding a degree one vertex, (b) adding a weak twin, or (c) adding a strong twin to a distance hereditary graph [1]. Thus, there exists in the deck D of G a distance-hereditary graph G^* such that G^* is obtained by removing a degree one vertex, a weak twin, or a strong twin. For every graph G' in D, we check if G' can be G^* by adding a degree one vertex, weak twin, strong twin. Adding a degree one vertex takes $O(1)$ time, and the number of ways of adding is $O(|V|)$. Adding a weak twin and a strong twin takes $O(|E|)$ time, and the number of ways of adding is $O(|V|)$. However the total cost is $O(|V| + |E|)$, since every edge copied exactly twice. For all the candidate of G^*, we use the $O(|V|^2(|V|+|E|))$ time DECK CHECKING algorithm. Therefore total time complexity is $O(|V| \cdot (|V| + (|V| + |E|) + |V| \cdot |V|^2(|V| + |E|)))$=$O(|V|^4(|V| + |E|))$.

4. Concluding Remarks

Since we can use PREIMAGE CONSTRUCTION algorithms for LEGITIMATE DECK and PREIMAGE COUNTING, we also have the LEGITIMATE DECK and PREIMAGE COUNTING algorithms running in the same time complexity for permutation (distance-hereditary) graphs. These results do not help directly the proofs of the graph reconstruction conjecture on permutation graphs. The conjecture on permutation (distance-hereditary) graphs still remains to be open.

We presented a polynomial time algorithm for PREIMAGE CONSTRUCTION on permutation graphs and distance-hereditary graphs. PREIMAGE CONSTRUCTION on interval graphs is solvable in polynomial time [12]. Kratsch and Hemaspaandra showed that PREIMAGE CONSTRUCTION on graph class C is GI-hard if the graph isomorphism is GI-hard on C [13]. Remaining famous graph class that we can find in [4] on which graph isomorphism is not GI-hard is circular-arc graphs (of course there are other non-GI-hard classes such as threshold graphs. However we mention here higher classes in the hierarchy of the inclusion relation). PREIMAGE CONSTRUCTION on circular-arc graphs may be a challenging problem. Another interesting graph class is circle graphs. Ma and Spinrad showed that a circle graph G has a unique representation if G is a prime with respect to split decomposition [14]. Split decomposition is a generalization of modular decomposition. Therefore it may be possible that PREIMAGE CONSTRUCTION on circle graphs is solvable in polynomial time in a similar way described in this paper. Circle graphs contain permutation graphs and distance-hereditary graphs.

References

[1] H. Bandelt and H.M. Mulder, "Distance-hereditary graphs," J. Combinatorial Theory, Series B, vol.41, pp.182–208, 1986.
[2] B. Bollobás, "Almost every graph has reconstruction number three," J. Graph Theory, vol.14, pp.1–4, 1990.
[3] J.A. Bondy, "A graph reconstructor's manual," in Surveys in Combinatorics, London Mathematical Society Lecture Note Series, vol.166, pp.221–252, 1991.
[4] A. Brandstädt and J.P. Spinrad, "Graph classes: a survey," SIAM, 1999.

[5] E. Dahlhaus, J. Gustedt, and R.M. McConnell, "Efficient and practical algorithms for sequential modular decomposition," J. Algorithms, vol.41, pp.360–387, 2001.
[6] S. Even, Algorithmic Combinatorics, Macmillan, New York, 1973.
[7] T. Gallai, "Transitiv orientierbare graphen," Acta Mathematica Hungarica, vol.18, pp.25–66, 1967.
[8] D.L. Greenwell and R.L. Hemminger, "Reconstructing the n-connected components of a graph," Aequationes Mathematicae, vol.9, pp.19–22, 1973.
[9] F. Harary, "A survey of the reconstruction conjecture," in Graphs and Combinatorics, Lecture Notes in Mathematics, vol.406, pp.18–28, 1974.
[10] E. Hemaspaandra, L. Hemaspaandra, S. Radziszowski, and R. Tripathi, "Complexity results in graph reconstruction," Discrete Appl. Math., vol.152, pp.103–118, 2007.
[11] P.J. Kelly, "A congruence theorem for trees," Pacific Journal of Mathematics, vol.7, pp.961–968, 1957.
[12] M. Kiyomi, T. Saitoh, and R. Uehara, "Reconstruction of interval graphs," Theor. Comput. Sci., vol.411, pp.3859–3866, 2010.
[13] D. Kratsch and L.A. Hemaspaandra, "On the complexity of graph reconstruction," Mathematical Systems Theory, vol.27, pp.257–273, 1994.
[14] T.H. Ma and J.P. Spinrad, "An O(n^2) algorithm for undirected split decomposition," J. Algorithms, vol.16, pp.145–160, 1994.
[15] B.D. McKay, "Small graphs are reconstructible," Australasian Journal of Combinatorics, vol.15, pp.123–126, 1997.
[16] S. Nakano, R. Uehara, and T. Uno, "A new approach to graph recognition and applications to distance-hereditary graphs," J. Computer Science and Technology, vol.24, pp.517–533, 2009.
[17] J.H. Schmerl and W.T. Trotter, "Critically indecomposable partially ordered sets, graphs, tournaments and other binary relational structures," Discrete Mathematics, vol.113, pp.191–205, 1993.
[18] J. Spinrad and J. Valdes, "Recognition and isomorphism of two-dimensional partial orders," in ICALP 1983, Lect. Notes Comput. Sci., vol.154, pp.676–686, 1983.
[19] J.P. Spinrad, Efficient Graph Representations, AMS, 2003.
[20] W.T. Tutte, "On dichromatic polynomials," J. Combinatorial Theory, vol.2, pp.310–320, 1967.
[21] M. von Rimscha, "Reconstructibility and perfect graphs," Discrete Mathematics, vol.47, pp.283–291, 1983.

Masashi Kiyomi recieved B.E. and M.E. degrees from University of Tokyo in 2000, and 2002, respectively. He recieved Ph.D. degree from National Institute of Informatics (in Japan), in 2006. He was an assistant professor at School of Information Science, Japan Advanced Institute of Science and Technology during 2006-2012. He is an associate professor at International College of Arts and Sciences, Yokohama City University.

Toshiki Saitoh recieved B.S.E. degree from Shimane University in 2005, and M.S. and Ph.D. degrees from Japan Advanced Institute of Science and Technology in 2007 and 2010, respectively. He was a researcher of ERATO MINATO Discrete Structure Manipulation System Project, Japan Technology and Science Agency, during 2010-2012. He is an assistant professor at Department of Electrical and Electronic Engineering, Graduate School of Engineering, Kobe University.

Ryuhei Uehara recieved B.E., M.E., and Ph.D. degrees from the University of Electro-Communications, Japan, in 1989, 1991, and 1998, respectively. He was a researcher in CANON Inc. during 1991-1993. In 1993, he joined Tokyo Woman's Christian University as an assistant professor. He was a lecturer during 1998-2001, and an associate professor during 2001-2004 at Komazawa University. He moved to Japan Advanced Institute of Science and Technology (JAIST) in 2004 as an associate professor, and since 2011, He has been a professor in School of Information Science. His research interests include computational complexity, algorithms, and data structures, especially, randomized algorithms, approximation algorithms, graph algorithms, and algorithmic graph theory. He is a member of EATCS, ACM, and IEEE.